

Module- 4Solved Question and answers on Python Programming.

Q] What is class? How do we define a class in Python? How to instantiate the class & how class members are accessed?

[PM-] [July/Aug 2022] V.V.V Imp

Ans] A class in python is like a blueprint for creating objects. It defines the properties (variables) and behaviors (functions) that objects created from the class will have.

```
class MyClass:
    greeting = "Hello!"
    def say_hello(self):
        return "Hello from the class!"
```

→ This is a property (class attribute)

→ This is a method (class function)

How to create an object from a class :-

You create an object by calling the class name like a function

my_object = MyClass() → Create an object of MyClass.

How to access class members (properties & methods)

Use the dot (.) operator to access the class's properties and methods through an object.

print(my_object.greeting) → output: Hello!

→ Accessing the property.

print(my_object.say_hello()) → Output: Hello from the class!

→ Accessing the method.

Example :

```
class Person:
    species = "Human"
    def say_name(self, name):
        return "My name is {name}."
```

person1 = Person() → Instantiate the class

print (person1.species) → Access the property.

Output : Human.

print (person1.say_name("Alice")) Output : My name is Alice.

Conclusion :

Define a class : Use the class keyword.

Create an object : Use object_name = ClassName()

Access properties : Use object_name.property.

Call methods : Use object_name.method (arguments)

3] Explain the role of Assertions in Python with a suitable program [6M.] V.V.V Imp

Ans

Refer pg no 9, 13 - 15//

Example :-

```
# initializing number
a = 4
b = 0
# using assert to check
for 0
```

```
print ("The value of a/b is ")
assert b != 0, "Zero Division Error"
print (a/b)
```

Output :-

AssertionError: Zero Division Error.

3] Explain the functions with examples :

- (i) `shutil.copytree()`
 (ii) `shutil.move()`
 (iii) `shutil.rmtree()`

[V.V.V Imp]

[6M-]

Ans Refer pg no 1-3

4] Develop a python program to traverse the current directory by listing sub folders & files. [6M-]

Ans

`import os`

`for foldername, subfolder, files in os.walk ('/home/secabiet/test'):`

`print ('current folder is ' + foldername)`

`for subf in subfolder:`

`print ('subfolder of ' + foldername + ':' + subf)`

`for filename in files:`

`print ('files inside of ' + foldername + ':' + filename)`

Output :-

current folder is /home/secabiet/test

subfolder of : /home/secabiet/test:test1

file inside of : /home/secabiet/test:a.py

current folder is /home/secabiet/test/test1.

Refer pg no 3-4 //

5] Explain the support for Logging with logging module in Python
[-8M-]

Ans page no 15 - 17 //

6] List out the benefits of compressing file ? Also explain reading a zip file [-8M-] V.V.V Imp

Refer page no 5-6

7] Develop a program with a function named DivExp which takes two parameters a, b & returns a value c ($C = a/b$). Write a suitable assertion for $a > 0$ in function DivExp and raise an exception for, when $b = 0$. Develop a suitable program which reads two values from the console & calls a function DivExp. [-8M-]

Ans

```
def DivExp(a, b):
    # AssertionError with error_message.
    assert a > 0, "Value of a should be greater than 0"
    if b == 0:
        raise Exception('Denominator should be greater than 0')
    c = a/b
    return(c)

a = int(input("Enter 1st no: "))
b = int(input("Enter 2nd no: "))
res = DivExp(a, b)
print("The result is: ", res).
```

denominator can't be 0

8] Write a note on Raising Exceptions . v.v.v. Imp

Ans Refer pg no 9 to 11

9] Write a note on Idle's Debugger.

Ans Refer pg no 17 .

10] Explain the following file operations in Python with suitable example .

- (i) Copying files & folders.
- (ii) Moving files & folders.
- (iii) Permanently deleting files & folders. [GM-]

Ans (i) Copying Files & Folders :-

To copy files & folders in Python, you can use the shutil module, specifically shutil.copy() & shutil.copytree()

Eg:- `import shutil`
`shutil.copy ("source_file.txt", "destination_file.txt")` → copy a file.
`shutil.copytree ("source_folder", "destination_folder")`
 → copy an entire folder.

`shutil.copy()` → copies a file from one location to another.
`shutil.copytree()` → copies an entire folder (includes its files and subfolders) to a new location.

(ii) Moving Files & Folders :-

For moving files & folders, use `shutil.move()`. This can move files

or directories to a new location.

Eg:-

```
import shutil
```

```
shutil.move("source_file.txt", "destination_file.txt").
```

```
shutil.move("source_folder", "destination_folder")
```

→ Move a file

→ Move a folder.

`shutil.move()` → Moves files or folders from one location to another. It can also rename files or folders while moving.

(iii) Permanently Deleting Files & Folders :-

↳ To delete file & folders permanently, you can use:

- `os.remove()` for files.
- `shutil.rmtree()` for folders

Eg

```
import os
```

```
import shutil
```

```
os.remove("file_to_delete.txt")
```

```
shutil.rmtree("folder_to_delete")
```

→ Delete a file

→ Delete an entire folder.

`os.remove()` → Deletes a single file

`shutil.rmtree()` → Deletes a folder & all its contents (files and subfolders) permanently.

Conclusion :-

1. Copying → Use `shutil.copy()` or `shutil.copytree()`
2. Moving → Use `shutil.move()`
3. Deleting → Use `os.remove()` for files & `shutil.rmtree()` for folders.